MoStBioDat – Molecular and Structural Bioinformatics Database

Andrzej Bak[*,1], Jaroslaw Polanski[1], Thomas Stockner[2] and Agata Kurczyk[1]

[1]*Department of Organic Chemistry, Institute of Chemistry, University of Silesia, PL-40-006 Katowice, Poland*

[2]*Austrian Research Centers, Department of Bioresources, 2444 Seibersdorf, Austria*

Abstract: Computer simulations play a crucial role in contemporary chemical investigations, generating enormous amounts of data. The constraint of sharing data and results is regarded as a major impediment in drug discovery. Among the steepest barriers to overcome in the high throughput screening studies is the limited number of suitable, freely accessible repositories for storing drug and drug target data. By offering a uniform data storage and retrieval mechanism, various data might be compared and exchanged easily. This paper presents the stages of the MoStBioDat software platform development, originally designed for the efficient storage, management and access of SDF and PDB data. The detailed architecture and software implementation of this project are described, indicating also the disadvantages of the solutions chosen. The current implementation of the first prototype is written in Python, an open-source, high-level, object-oriented scripting language. The modular architecture of the package enables future extension with the necessary functionalities. The main objective of the MoStBioDat is to serve as an alternative, extensible open-source database derived partly from SDF and PDB files.

Keywords: SDF, PDB, SMILES, HTS, relational database, ligand, macromolecule, python, chemoinformatics.

1. INTRODUCTION

Recent technological progress being made in domain of computational chemistry is constantly increasing the number of chemoinformatics/bioinformatics resources combining detailed drug data with the comprehensive drug target information [1]. Moreover, the computationally demanding molecular simulations, which form an increasingly important component of the current investigations, can generate enormous amount of data. These data might be examined by a variety of methods which complement the experimental studies. The aggregation and organization of dataset of chemical information enables for massive *in silico* processing in contrast to the traditional way of handling data in the current molecular design which is much less efficient and user-friendly. It is observed that the development of chemoinformatics has been hampered by the lack of the unified standards for data storage, management and exchange [2]. Consequently, the missing ability of sharing and comparing the simulation data is among the steepest barriers to overcome in this scientific area. As a result, the tools and techniques for organizing and intelligently mining this information are highly desirable. Implementing, handling and searching chemical libraries or databases in so-called virtual screening constitute a rapidly growing field in drug discovery which has benefited considerably from improvements in computer technology [3]. The virtual screening should not be primarily focused on gathering data but also on developing efficient and robust tools for assessing and predicting molecular properties. Offering a uniform data storage and extraction mechanism with the extensive array of tools for structural similarity measures and pattern matching is essential to facilitate the drug discovery process.

*Address correspondence to this author at the Department of Organic Chemistry, Institute of Chemistry, University of Silesia, PL-40-006 Katowice, Poland; Tel: 0048 032 3591399; Fax: 0048 032 2599979;
E-mail: abak@us.edu.pl

In the high throughput screening (HTS) approach, two general research strategies have been developed. The first one, based on the knowledge of the target geometry relies on searching a bioeffector as a complementary fulfillment of the receptor structure (*RD – receptor dependent*) [4], whereas the second one is looking for the analogies in the ensemble of the active ligands with the creation of the spatial maps of interactions so-called pharmacophore (*RI – receptor independent*) [5-8]. The structure-based database screening has recently become a common and efficient technique in an early stages of the rational drug design. Moreover, molecular property prediction can significantly decrease the cost of drug discovery [9, 10]. In spite of the increasing number of freely available web databases, only a few repositories combining the knowledge of small molecules (ligand) and their corresponding drug targets (macromolecule) are currently available [11]. Unfortunately, the HTS technique in many cases requires experience and expert knowledge of the database querying language (SQL), so it remains out of reach for many medically oriented investigators. In an effort to make the virtual screening more accessible to the scientific community, the Molecular and Structural Bioinformatics Database (*MoStBioDat*) project has been established as a management platform for an efficient storage, access and exchange of the biomolecular data. It could potentially serve as a dual purpose storage environment integrated with a database management system (DBMS) to explore 3D drug-target interactions or compare and measure the structural similarities between chemical structures.

2. PROJECT CONCEPT

The main objective of this development was to create a software platform offering the way to deposit data in a unified format and avoiding unnecessary data replication, while maintaining the high standards of data integrity and reliability. Additionally, this knowledge based system for data exploration provides a consistent environment for data

1386-2073/10 $55.00+.00

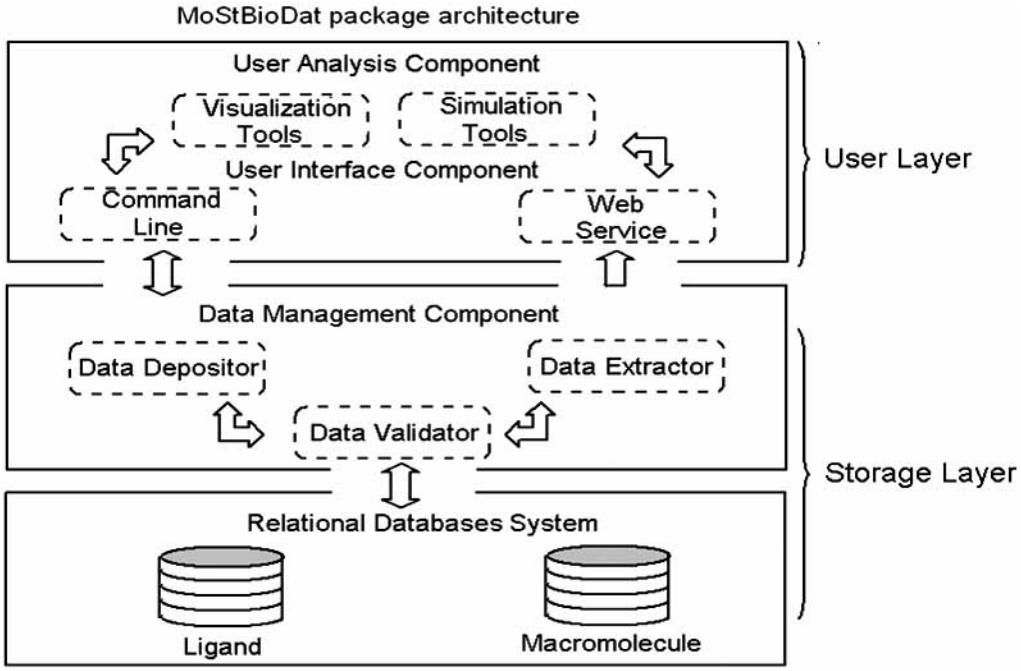

Fig. (1). The concept of the MoStBioDat architecture.

analysis. Practically, the entire system is based on the client-server architecture, although the current prototype was implemented and tested where both the applications and database servers were running at a single PC location. The brief overview of the MoStBioDat system is given in Fig. (**1**). Conceptually, the system architecture currently encompasses the following components: *Storag*e and *User Layer*, respectively. At the stage of designing this system, we have compared the architecture with the BioSimGrid project, which has a similar structural layout [12, 13].

The *Storage Layer* is in charge of storing and preserving data with simple methods for data querying and retrieving. Following the contemporary trends in the database technology, the pragmatic relational approach has been applied, because of its availability, simplicity and flexibility [14]. Therefore, the database system has been designed to organize data relationally with the parent-child key relationships what enables an efficient management of the stored datasets with the open-source MySQL as a database server [15]. The relational database system is composed of two major ingredients: *Ligand* and *Macromolecule*, respectively. The underlying complexity of parsing, validating, storing and extracting data is hidden in the middleware called *Data Management Component*, which provides an abstract layer with sets of services responsible for making data transparently available. The post-processing component in the *User Layer* called *User Interface Component* offers two modes of browsing the databases. Expert users have access to datasets with fully programmable Python [16] command line which is helpful for more specific analysis, whereas novice users can apply the graphical user interface (GUI) – not implemented yet in the current prototype. A set of pretty mature python libraries and programs, for instance MMTK, VMD or R, are freely available with all necessary simulation and visualization tools [17-19].

2.1. Input Data

Small molecules (ligands) with at most a few dozen atoms play a fundamental role in organic chemistry being used as a building blocks in combinatorial chemistry as well as a molecular probes for analyzing biological systems in drug discovery [20]. Several databases of ligands are accessible publicly combining multiple molecule representations with numeric and alphanumeric metadata [21]. The SDF (structure-data file) is probably the most widely known standardized annotation format containing the structural information and associated tagging items for large-scale data transfer between databases [22, 23]. To avoid common problems with calculating the correct protonation, tautomeric and 3D conformational states for datasets, the ZINC “drug-like” SDF ensemble has been directly used as a basic source of data for the *Ligand* part [24]. The current version of the *Ligand* database contains approximately 5 million compounds.

The receptor dependent simulations (RD) have contributed significantly to investigations of drug-target interactions, especially in the case of systems difficult to probe experimentally [25]. Macromolecules are hierarchically conceptualized as a functional ensemble, molecule, polymer chains, domains, secondary structure elements, residues, side chains, small molecules, and atoms [26]. The Protein Data Bank (PDB) established at Brookhaven National Laboratories (BNL) is the major archive containing 3D structures of proteins, nucleic acids and other biological macromolecules experimentally determined using X-ray crystallography, NMR or electron microscopy techniques [27-29]. Although the PDB format, which was originally designed to be human readable, has evolved during the last decades into several PDB dialects, it is still the main flat-file format for macromolecular models manipulations used by many computer programs [30, 31]. The representation of atomic coordinates and related

information as well as the description of the experiment details of structure determination are encoded with a record tag name followed by the individual items of data [32]. The *Macromolecule* database serves as an alternative, expandable storage repository to manipulate the PDB data.

2.2. SMILES Code

Coding of chemical structure is an essential issue for computerized high-speed processing of chemical information, mainly used in the database screening. Among several coding algorithms, the SMILES (Simplified Molecular Input Line Entry System) notation used to represent a molecular structure by a linear string of symbols is particularly popular [33, 34]. A molecule is denoted as a two-dimensional, undirected graph with the optional chirality indications where each atom/node is enumerated by a numerical label given on the basis of its topology (invariant node property and connectivity) [35]. Unfortunately, there is no natural order of nodes in a molecular graph resulting in a unique string representation independent of the input order of atoms. As a result, there might be a few generic SMILES representing a given structure which is highly undesirable in molecular comparisons. Advanced canonicalization algorithms have been developed to generate a tree representation of the molecular graph resulting in one special generic SMILES among many valid possibilities – unique/canonical SMILES notation [36]. The problem of the similarity estimation in a set of canonically ranked molecules is reduced to measure the similarities of ASCII-coded strings. The indexation of the SMILES notation speeds up significantly the database engine screening operations.

2.3. Programming Language

The Python as a free, open-source, platform-independent and very-high-level (VHLL) object-oriented programming language has been chosen to implement the first prototype of the MoStBioDat package because of its simplicity, flexibility and dynamic extensibility [37, 38]. The modular architecture of this interpreted language enables the integration of the standard libraries with many mature, well-designed external modules for convenient usage to avoid re-coding of the common task (OEChem, Pybel, MySQLdb, ForgetSQL library) [39-41]. The combination of the expressive scripting language with some functionalities of the high-performance chemoinformatics toolkits creates a unique mixture to manipulate chemical data, thus Python is also preferred as a post-processing environment.

The physical implementation of the conceptual database design has been conducted with the leading, open-source, multi-threaded and multi-user robust MySQL system. Be aware of some speed limitations of MySQL and Python scripting language in comparison with some commercially available solutions or compiled languages. The detailed description of SDF/PDB file format, SMILES notation, Python programming language and MySQL database server is beyond the scope of this paper.

3. RESULTS AND DISCUSSION

3.1. Database Structure

To maintain an internally coherent archive, the relational approach has been applied to design the basic database concept. Following some logical SDF/PDB flat-file organizations, the data items are mapped into the relational tables with the integer identification number (Id) as a primary key for rapid indexing and enforcement of uniqueness. The uniquely identified table data may be easily parsed using binary searching procedures performed directly within the computer RAM memory. Assuming the proper design of the mutual table relationships, centralized checking of the foreign key constraints enforces the referential data consistency and integrity. In practice, the parent-child relationships are specified by lines matching the primary key of the parent table with the suffixed (fk) column of the child table pointing out the kind of relationship.

The *Ligand* database system, shown in Fig. (**2**), contains a set of the primary tables for supporting the multiple molecule representations including one-dimensional description with the SMILES code (*ChemComp* table), protonation state (*ProtStat* table) and conformational sampling (*ConfStat* table), respectively. The expandable property specification table (*PropDef*) depicts a molecule with the definitions of the alphanumeric and numeric data. Additionally, the supportive set of subtables for each of the primary tables storing the specific values of the defined properties has been established. The unique representations of the particular chemical compound has been ensured by calculating the canonical SMILES representations with the OpenEye's OEChem library [39].

The *Macromolecule* database, introduced in Fig. (**3**), has been designed in order to be able to store and retrieve the original PDB flat file. The main biopolymer concept assumes that macromolecule consists of the well defined building blocks (residues) which are connected according to the underlying sequence. The topology of the biomolecular system is stored in the tables: *Entry*, *Molecule*, *Residue*, *HetMol* etc., which generally reflects the conventional PDB file structure and data hierarchy. The central *Entry* table brings together the structural data and metadata. The database system stores some metadata in a set of supportive tables, for instance, *DBRef*, *JournalDat* and *MatrixDat*. The total system, including *Ligand* and *Macromolecule*, is integrated to create a consistent storage environment for data derived from PDB flat-files. The entire database system including binary log files occupies nearly 350 GB of the hard drive space. From various database design procedures, we have chosen the relational approach being aware of the fact that it is not the ideal solution for complicated query performance, especially where data are spread out over many different tables.

3.2. Package Architecture

The intention of this short section is to present roughly the range of software components and its corresponding

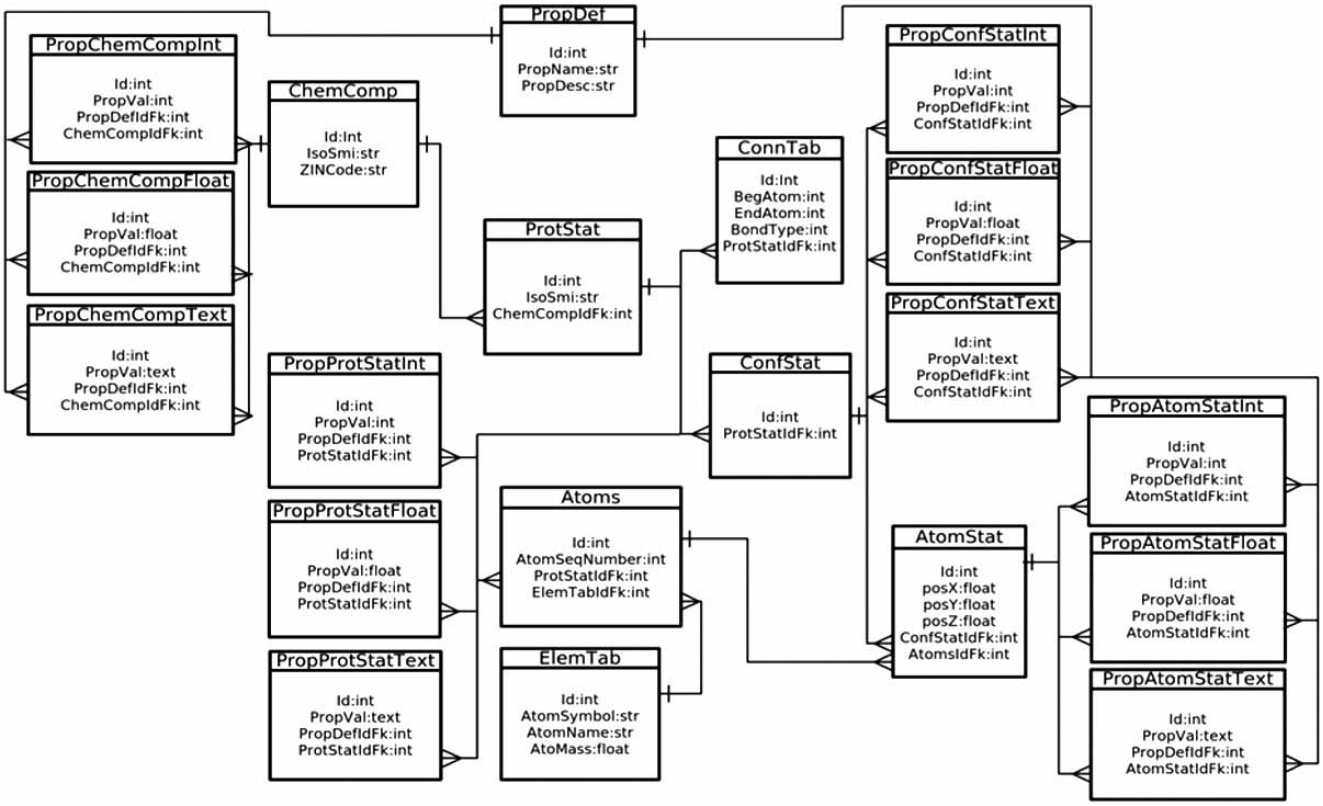

Fig. (2). The Ligand database scheme.

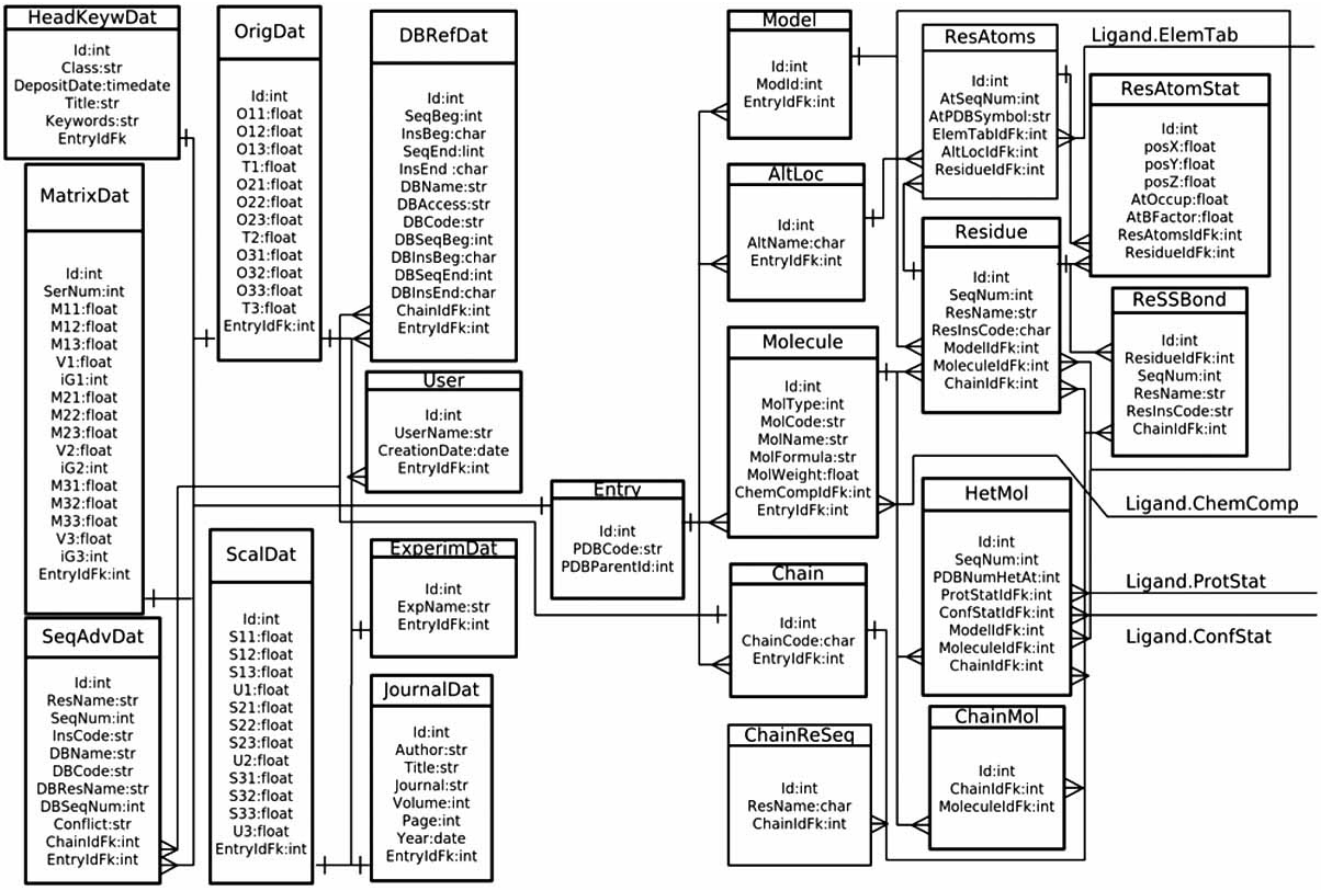

Fig. (3). The Macromolecule database scheme.

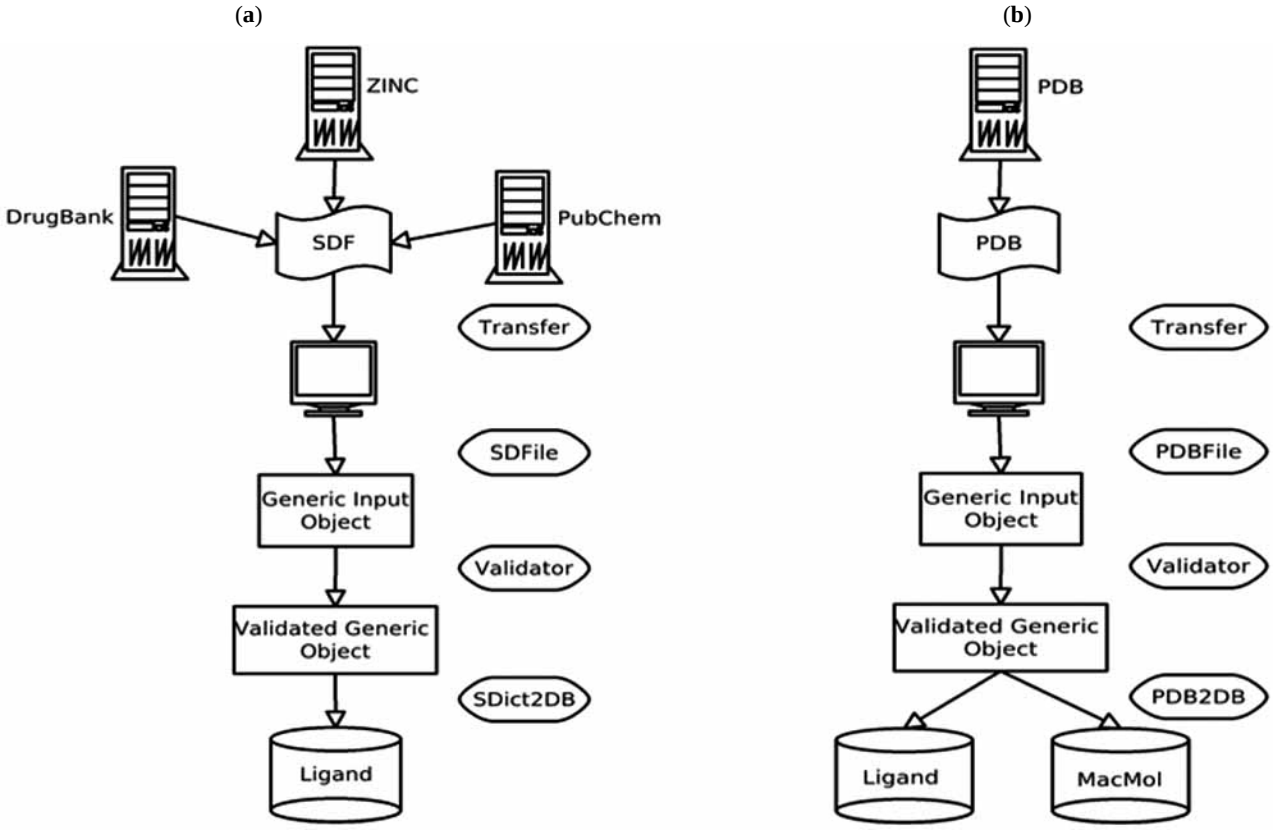

Fig. (4). Data storage procedure for data taken from the SDF (**a**) and PDB (**b**) files, respectively.

functionality composing the software toolkit. A brief outline of the object-oriented package design is given in Table **1**. The modular approach applied in the implemented components brings together a basic set of functionalities for data manipulation, resulting in the extensible software application. The package consists of a set of modules, each performing a given task on data or database, providing software-based solutions for the semi-automated executions of *in-silico* protocols. The provided library of routines automates the complex task of building a structural object hierarchy from given data resources with the simple methods for accessing data transparently. The complexity and understanding of the data storage structure and data querying methods are hidden in the software component which creates a user-friendly data management environment.

Additionally, the package offers a simple and quick installation procedure for setting up the working system. The detailed description of the source code and module functionality with many examples of usage are accessible *via* a web browser in the MoStBioDat/Docs directory. The basic hardware and software requirements, short instruction of the installation and some test files are distributed within the package MoStBioDat/Data directory. The MoStBioDat source code is currently released under the GPL 3 license.

3.3. Examples of Usage

The detailed functional description of the particular modules seems to be aimless, therefore some typical scenarios of the data manipulation procedures, for instance the data deposition and extraction, are shown in Figs. (**4**, **5**), respectively. To download specified SDF files (Fig. **4a**) from a remote site (ZINC, PubChem, DrugBank), the *Transfer* module containing the FTP and SFTP tools can be applied. The *Deposition Component* consists of parsers (SDFile, PDBFile) based on the OEChem library to translate the simulation data into a generic input object. Then, this object is parsed through the *Validator* to check the consistency between data type and the database table column description. The entire process is finished when the positively validated generic object is deposited in the database. In case of the PDB files (Fig. **4b**), data can be divided into parts and imported to the *Ligand* and *Macromolecule* database, respectively.

The *Retrieval Component* queries the database to extract all needed and available information (Fig. **5**). This component abstracts from the low-level SQL syntax and is responsible for data gathering. Different ways of database query creation are offered using the predefined syntax (UserQuery module), the simple Python syntax (ForgetSQL*2*) [40] or the SQL syntax (SQLBuilder) [41]. Having released the query, optionally the cache memory can be consulted to check whether or not the specific query had been created previously. If not, then the database is questioned for the results. The limit on the number of queries held in the cache file is defined by the user. After crossing this limit, the compressed archive is created, and a new cache file is initialized. The query result is the Python array of objects being available for users with all pythonic manipulation tools for example the numpy package. Two modules (DB2SDF and DB2PDB) for a specific file format creation in a fully automated way are provided. The Python

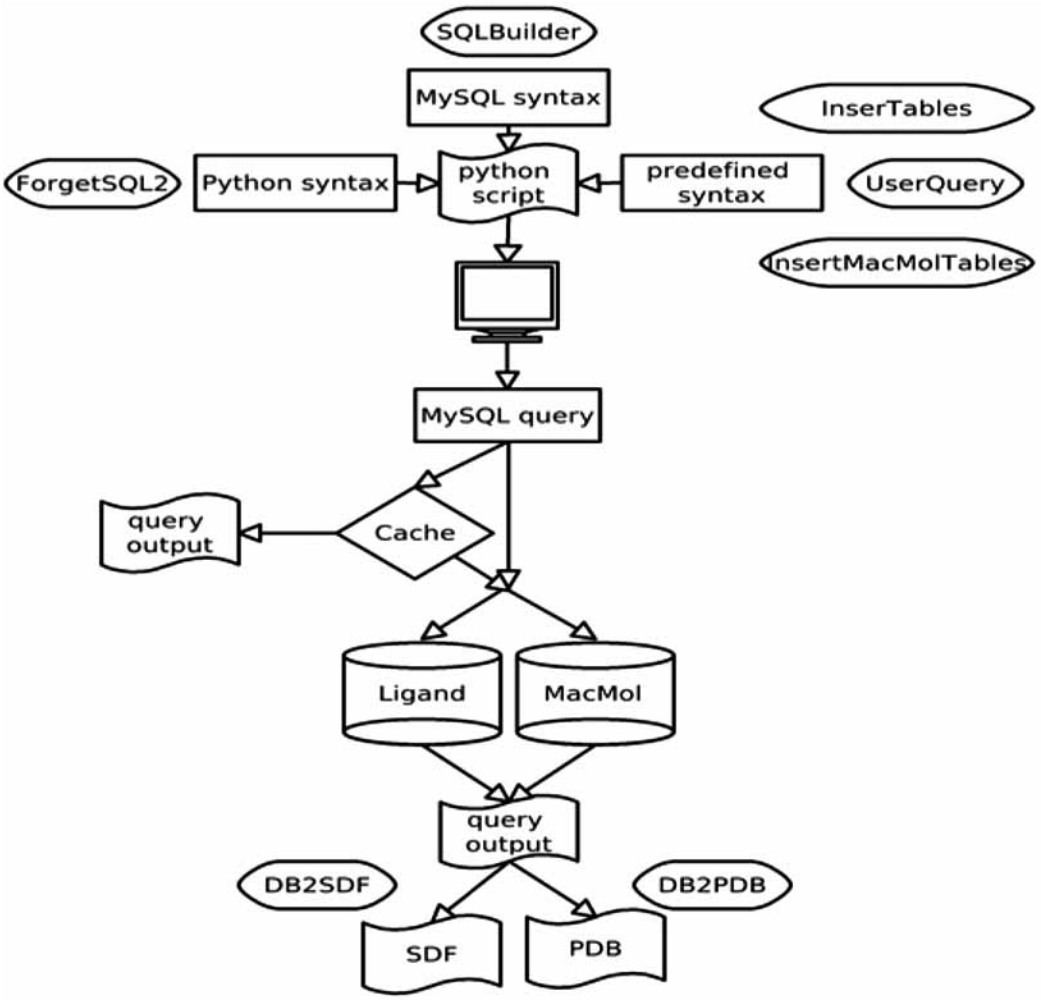

Fig. (5). Data retrieval procedure.

programming environment offers post-processing tools for data simulation and visualization.

3.4. Substructure Search

The rapid process of subsequent examination of a large number of molecules stored in a database combined with the calculation of the metric quantifying the similarity with a given pattern is one of the fundamental tasks for computers in chemistry [42, 43]. Typically, the screening procedure takes into account the entire structure or substructure as a search query for identifying the ensemble of compounds fulfilling the global similarity criteria. In practice, the similarity measures derived by comparing the presence and/or absence of features or the occurrences of substructures present in a molecule are based on abstract representations of certain structural features of a compound in the form of the fixed-size binary fingerprint vector [44].

Characterizing a chemical structure in the binary form integrated with the efficient bit-wise algorithms yield high-speed structural screening procedures, which in comparison with the precise, computationally demanding exact search results might produce a bit higher rate of false positives [45]. Different similarity measures and pattern matching procedures have been developed for the molecular fingerprints as filtering methods identifying or eliminating drug-likeness of molecules. Among the most commonly used quantitative metric for estimating structural similarities is the Tanimoto [46, 47] coefficient defining the proportion of substructures in common between two molecules expressed by the following formula:

$$T(x,y) = \frac{n_{xy}}{(n_x + n_y - n_{xy})} \quad (1)$$

where:

n_{xy} - the number of bits set into 1 shared in the fingerprint of molecule x i y

n_x - the number of bits set into 1 in the molecule x

n_y - the number of bits set into 1 in the molecule y

The ratio approaching 0.7 or more indicates that compounds being compared are fairly structurally similar. Conceptually, the procedure of finding a particular pattern in a molecule might be interpreted as string regular expressions describing the search criteria. All compounds which share a common substructure might be identified using the straightforward extension of SMILES notation – the SMARTS pattern [46].

Table 1. The Set of Modules within the MoStBioDat Package

Module Name	Module Description
Transfer	FTP and SFTP tools
Log	logging module
Data	database installation package and post-installation test procedures
Database	
Connect	database connection methods
ForgetSQL2	internal ForgetSQL2 module
ImportData	
DB2Data	database retrieval module
Data2DB	database import module
Validator	database validation module
Query	predefined user database queries
Scheme	database scheme creation module (HTML)
SubStructSearch	substructure search methods

It is assumed that the largest common component which appears in the structurally related drugs might determinate their biological activity. The Maximum Common Substructure (MCS) approach is an alternative method of pattern matching which provides a similarity score for a pair of structures used as a metric for ranking the molecular similarity [48, 49]. The conversion of the MCS-based procedure into the CPU intensive maximum clique detection problem makes it impossible to be applicable in high-speed database screening.

The practical observation of some molecular properties important for drug's pharmacokinetics (ADME) in the human organism led to formulate the quantitative filter called Lipinski's Rule of Five (RO5) [50, 51]. The property space was restricted to the range of values defined by the octanol/water partition coefficient (ClogP≤5), the molecular weight (MW≤500), the number of hydrogen bond donors (HBD≤5), and the number of hydrogen bond acceptors (HBA ≤10), respectively. Roughly speaking, the violation of the above conditions might discriminate between prospective drugs and non-drugs, but RO5 does not represent the precise rules sufficient for druglikeness [1]. Taking into account more restrictive conditions (MW≤460, -4≤ClogP≤4.2, HBD≤5, HBA≤9) the leadlikeness criteria have been established to identify drug prototypes, which are optimized before obtaining the drug candidate status [52]. The leadlike-based strategy is also expected to be applicable for database sampling as a integral part of the continual enrichment of the HTS procedure.

The OpenBabel's [53] and OpenEye's [54] functions have been applied to generate and compare the molecular fingerprints and the SMARTS matching in the *SubstructureSearch* module (TanimotoSearch, SMART Search, QuerySearch, MCSearch, CliqueSearch, RO5 Search). The possible application of the system for the analysis of molecular diversity was described elsewhere [55].

3.5. Distribution

The brief description of the current development status with the latest released version of the package to download and the generated documentation are available under the following web address: http://www.chemoinformatyka.us.edu.pl/mostbiodat/. Under this address, the whole system can be downloaded to be installed locally on the user machine. The system can be installed on the linux system.

3.6. Future Developments

The provided MoStBioDat package is still being developed and refined. In particular, the web database access with the graphical user interface (GUI) is urgently needed to facilitate the process of query creation and data extraction. Additionally, attention should be paid to the evaluation of the strengths and weaknesses of the current prototype with the complex simulations procedures (MD).

4. CONCLUSIONS

The conceptual design presented above and practical software implementation of the MoStBioDat project should serve as an alternative, extensible management platform for data derived from the SDF and PDB flat-files. It provides the abstract layer with methods for data manipulation with the minimum efforts to install the running system. The relational database system maintains data in an organized form that reduces or partly eliminates data redundancy and enables efficient data management. The modular architecture of the package makes it possible to integrate completely new tools for a novel analysis procedures potentially needed in the future. It is believed that the MoStBioDat package will be a useful and user-friendly environment for database screening. The major advantage of this platform is the possibility of being installed locally, which makes the main difference to

the majority of platforms available. This clearly improves the efficiency of data manipulation during the massive calculations needed by the current virtual screening methods. Moreover, the user can personalize the hardware/software environment and data to import with a set of tools for storage, access and exchange of biological data. The modular architecture of Python package enables also any extensions, if needed by user.

ACKNOWLEDGEMENTS

We would like to acknowledge the OpenEye and OpenBabel Scientific Software for the free software academic license. Dr. Andrzej Bak thanks: Foundation for Polish Science for his individual grant, Dr. Stian Soiland-Reyes for ForgetSQL2 python module, Dr. Ian Bicking for SQLBuilder python module, Dr. Beata Zawisza and Dr. Michael Jakusch for their valuable remarks. This work was partially financed by Polish Ministry of Science grant N405 178735.

REFERENCES

[1] Olah, M.; Bologa, C.; Oprea, S.T. Strategies for compound selection. *Curr. Drug Discov. Tech.*, **2004**, *1*, 211-220.

[2] Jónsdóttir, S.; Jørgensen, F.; Søren, B. Prediction methods and databases within chemoinformatics: emphasis on drugs and drug candidates. *Bioinformatics*, **2005**, *21*, 2145-2160.

[3] Burbaum, J.; Sigal, N. New technologies for high throughput screening. *Curr. Opin. Chem. Biol.*, **1997**, *1*, 72-78.

[4] Santos-Filho, O.; Hopfinger, A. Structure-based QSAR analysis of a set of 4-hydroxy-5,6-dihydropyrones as inhibitors of HIV-1 protease: an application of the receptor dependent (RD) 4D-QSAR formalism. *J. Chem. Inf. Model.*, **2006**, 345-354.

[5] Polanski, J. Self-organizing neural networks for pharmacophore mapping, *Adv. Drug. Deliver. Rev.*, **2003**, *55*, 1149-1162.

[6] Bak, A.; Polanski, J. The 4D-QSAR study on anti-HIV HEPT analogues. *Bioorg. Med. Chem.*, **2006**, *14*, 273-279.

[7] Bak, A.; Polanski, J. Modeling robust QSAR 3: SOM-4D-QSAR with iterative variable elimination IVE-PLS: application to steroid, azo dye and benzoic acid series. *J. Chem. Inf. Model.*, **2007**, *47*, 1469-1480.

[8] Polanski, J.; Bak, A.; Gieleciak, R.; Magdziarz, T. Modeling robust QSAR *J. Chem. Inf. Model.*, **2006**, *46*, 2310-2318.

[9] Wishart, D.; Knox, C.; Guo, A.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Wollsey, J. DrugBank: a comprehensive resource for *in silico* drug discovery and exploration. *Nucleic Acids Res.*, **2006**, *34*, 668-672.

[10] Chen, J.; Swamidass, S.; Dou, Y.; Bruand, J.; Baldi, P. ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinformatics*. **2005**, *21*, 4133-4139.

[11] *http://www.ebi.ac.uk/pdbe/docs/References.html*

[12] Wu, B.; Tai, K.; Murdock, S.; Ng, M.; Johnston, S.; Fangohr, H.; Jeffreys, P.; Cox, S.; Essex, J.; Sansom, M. BioSimGrid: a distributed database for biomolecular simulations. In: *Proceeings of the UK e-Science All Hands Meeting*, Cox, J., Ed.; Swindon, **2003**, 412-419.

[13] Ng, M.; Johnston, S.; Murdock, S.; Wu, B.; Tai, K.; Fangohr, H.; Cox, S.; Essex, J.; Sansom, M.; Jeffreys, P. Efficient data storage and analysis for generic biomolecular simulation data. In: *Proceedings of UK e-Science All Hands Meeting.* Cox, J., Ed.; Swindon, **2004**, 443-450.

[14] Dong, X.; Gilbert, E.; Guha, R.; Heiland, R.; Kim, J.; Pierce, E.; Fox, C.; Wild, J. Web service infrastructure for chemoinformatics. *J. Chem. Inf. Model.*, **2007**, *47*, 1303-1307.

[15] *http://www.mysql.com/*

[16] *http://www.python.org*

[17] *http://dirac.cnrs-orleans.fr/MMTK/*

[18] *http://www.ks.uiuc.edu/Research/vmd/*

[19] *http://www.r-project.org/*

[20] Swamidass, S.; Chen, J.; Bruand, J.; Phung, P.; Ralaivola, L.; Baldi, P. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, **2005**, *21*, 359-368.

[21] von Grotthuss, M.; Koczyk, G.; Pas, J.; Wyrwicz, L.; Rychlewski, L. Ligand Info small-molecule meta-database. *Comb. Chem. High Throughput Screen.*, **2004**, *7*, 757-761.

[22] Dalby, A.; Nourse, J.; Hounshell, W.; Gushurst, A.; Grier, D.; Leland, B.; Laufer, J. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *J. Chem. Inf. Comput. Sci.*, **1992**, *32*, 244-255.

[23] Hushurst, A.; Nourse, J.; Hounshell, W.; Leland, B.; Raich, D. The substance module: the representation, storage and searching of complex structures. *J. Chem. Inf. Comput. Sci.*, **1991**, *31*, 447-454.

[24] Irwin, J.; Shoichet, B. ZINC – A free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.*, **2005**, *45*, 177-182.

[25] Huang, N.; Jacobson, M. Physics-based methods for studying protein-ligand interactions. *Curr. Opin. Drug Discov. Devel.*, **2007**, *10*, 325-331.

[26] Westbrook, J.; Bourne, P. STAR/mmCIF: an ontology for macromolecular structure. *Bioinformatics*, **2000**, *16*, 159-168.

[27] Berman, H.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.; Wiessig, H.; Shindyalov, I.; Bourne, P. The Protein Data Bank. *Nucleic Acids Res.*, **2000**, *28*, 235-243.

[28] Sussman, J.; Lin, D.; Jiang, J.; Manning, N.; Prilusky, J.; Ritter, O.; Abola, E. Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr. D Biol. Crystallogr.*, **1998**, *54*, 1078-1084.

[29] Berman, H.; Henrick, K.; Nakamura, H.; Markley, J. The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res.*, **2007**, *35*, 301-303.

[30] Westbrook, J.; Ito, N.; Nakamura, H.; Henrick, K.; Berman, H. PDBML: the representation of archival macromolecular structure in XML. *Bioinformatics*, **2005**, *21*, 988-992.

[31] Deshpande, N.; Addess, K.; Bluhm, W.; Merino-Ott, J.; Wownsend-Merino, W.; Qing, Z.; Knezevich, C.; Xie, L.; Chen, L.; Feng, Z.; Green, K.R.; Flippen-Anderson, J.; Westbrook, J.; Berman, H.; Bourne, P. The RCSB Protein Data Bank: a redesigned query system and relational database based on the mmCIF schema. *Nucleic Acids Res.*, **2005**, *33*, 223-237.

[32] *http://mmcif.pdb.org/*

[33] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, **1998**, *28*, 31-36.

[34] Weininger, D.; Weininger, A.; Weeininger, J. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.*, **1989**, *29*, 97-101.

[35] Satoh, H.; Koshino, H.; Funatsu, K.; Nakata, T. Novel canonical coding method for representation of three-dimensional structures. *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 622-630.

[36] Koichi, S.; Iwata, S.; Uno, T.; Koshino, H.; Satoh, H. Algorithm for advanced canonical coding of planar chemical structures that considers stereochemical and symmeteric information. *J. Chem. Inf. Model.*, **2007**, *45*, 1734-1746.

[37] Martelli, A. Python in a Nutshell. 2nd ed. O'Reilly, **2006**, Sebastopol, CA 95472.

[38] *http://docs.python.org/tutorial/*

[39] OEChem – Python Theory Manual, OpenEye Scientific Software Inc., **2008**, Santa Fe, NM 87508.

[40] O'Boyle, N.; Morley, C.; Hutchison, R. Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chem. Cent. J.*, **2008**, *2*, 1-7.

[41] *http://soiland.no/i/src/forgetsql2/*

[42] *http://www.eyesopen.com/docs/html/api/*

[43] *http://www.sqlobject.org/SQLBuilder.html*

[44] Swamidass, S.; Baldi, P. Bounds and algorithms for fast exact searches of chemical fingerprints in linear and sub-linear time. *J. Chem. Inf. Model.*, **2007**, *47*, 302-317.

[45] Stahl, M.; Mauser, H. Database clustering with a combination of fingerprint and maximum common substructure methods. *J. Chem. Inf. Model.*, **2005**, *45*, 542-548.

[46] Daylight Theory Manual, http://www.daylight.com/dayhtml/doc/theory/theory.toc.html

[47] Flower, D. On the properties of bit-string based measures of chemical similarity. *J. Chem. Inf. Comput. Sci.*, **1998**, *38*, 379-386.

[48] Podolyan, Y.; Karypis, G. Common pharmacophore identification using Frequent Clique detection algorithm. *J. Chem. Inf. Model.*, **2009**, *49*, 13-21.

[49] Cao, Y.; Jiang, T.; Girke, T. A maximum common substructure-based algorithm for searching and predicting drug-like compounds, *Bioinformatics*, **2008**, *24*, 366-374.

[50] Lipinski, C.; Lombardo, F.; Dominy, B.; Feeney, P. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Del. Res.*, **2001**, *46*, 3-26.

[51] Oprea, T.; Davis, A.; Teague, S.; Leeson, P. Is there a difference between leads and drugs? A historical perspective. *J. Chem. Inf. Comput.*, **2001**, *s1*, 1308-1315.

[52] Hann, M.; Oprea, T. Pursuing the leadlikeness concept in pharmaceutical research. *Curr. Opin. Chem. Biol.*, **2004**, *8*, 255-263.

[53] *http://openbabel.org/wiki/Main_Page*

[54] *http://www.eyesopen.com/*

[55] Bak, A.; Polanski, J.; Kurczyk, A. The use of MoStBioDat for rapid screening of molecular diversity. *Molecules*, **2009**, *14*, 3436-3445.

Received: April 30, 2009 **Revised: July 22, 2009** **Accepted: July 27, 2009**